

Министерство образования и науки Российской Федерации
ФБГОУ ВО «Уральский государственный педагогический университет»
Институт математики, информатики и информационных технологий
Кафедра информационно-коммуникационных технологий в образовании

РАЗРАБОТКА УЧЕБНОГО КВЕСТА С ПОМОЩЬЮ ВИЗУАЛЬНОГО КОНСТРУКТОРА KODU GAME LAB

Выпускная квалификационная работа
по направлению подготовки 090302-Информационные системы и технологии

Работа допущена к защите
« ____ » _____ 2016 г.
Зав. кафедрой _____

Исполнитель: студент БС41z группы
института математики,
информатики и ИТ
Ивлев П.А.

Руководитель: ст. преподаватель
кафедры ИКТО
Старкова Л.Н.

Екатеринбург 2016

РЕФЕРАТ

ИВЛЕВ П.А. РАЗРАБОТКА УЧЕБНОГО КВЕСТА С ПОМОЩЬЮ ВИЗУАЛЬНОГО КОНСТРУКТОРА KODU GAME LAB, выпускная квалификационная работа: 49 стр., рис. 22, табл. 2, библиограф. 19 назв.

Ключевые слова: УЧЕБНЫЙ КВЕСТ, KODU GAME LAB.

Объект разработки – Учебный квест для уроков информатики по теме: «Файловые системы».

Цель работы – описать технологию разработки учебного квеста с использованием визуальной объектно-ориентированной среды программирования Kodu Game Lab.

В работе описаны результаты сравнения программных средств объектно-ориентированного программирования Scratch и Kodu Game Lab. Выделены преимущества визуального конструктора Kodu Game Lab. Описана подробная технология разработки учебного квеста. Разработан учебный квест в визуальном конструкторе Kodu Game Lab, состоящий из 4 уровней, по теме «Действия с файлами в операционной системе Windows», обеспечивающий в доступной и игровой форме обучение пользователя персонального компьютера взаимодействию с файлами в операционной системе Windows.

В учебный квест включены в качестве обязательных следующие части:

Первый этап предоставляет основную информацию, вводит ключевые понятия, а также содержит вопрос, над которыми будут размышлять учащиеся;

Второй этап содержит задание (цель, условия, проблема и пути ее решения). Это наиболее важная часть учебного квеста. Задание направляет учащихся на ряд конкретных действий на пути решения проблемы;

Третий этап – это обобщение результатов, подведение итогов (чему научились, какие навыки приобрели; возможны риторические вопросы или вопросы, мотивирующие дальнейшее исследование темы). Здесь подводится итог и поощряется рефлексия и дальнейшие исследования по данной теме.

Результаты работы были внедрены в учебный процесс Муниципального бюджетного общеобразовательного учреждения «Средняя общеобразовательная школа № 2» Асбестовского городского округа в 2016-2017 учебном году.

Оглавление

ВВЕДЕНИЕ	5
Глава 1. АНАЛИЗ KODU GAME LAB ДЛЯ РАЗРАБОТКИ УЧЕБНОГО КВЕСТА.....	9
1.1. Основные понятия квеста	9
1.2. Сравнительный анализ продуктов Scratch и Kodu Game Lab для разработки учебного квеста	13
1.3 Техническое задание	17
Глава 2. РАЗРАБОТКА УЧЕБНОГО КВЕСТА В ВИЗУАЛЬНОЙ СРЕДЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ KODU GAME LAB.....	19
2.1 Знакомство с визуальной объектно-ориентированной средой Kodu Game Lab	19
2.2 Создание учебного квеста в визуальной объектно-ориентированной среде Kodu Game Lab.....	32
2.3 Акт внедрения учебного квеста в образовательный процесс	44
ЗАКЛЮЧЕНИЕ.....	46
Список информационных источников	48

ВВЕДЕНИЕ

Стремительное развитие информационного общества, проявление и широкое распространение технологий мультимедиа, электронных информационных ресурсов, сетевых технологий позволяют использовать информационные технологии в качестве средства обучения, общения, воспитания, интеграции в мировое пространство. Совокупность традиционных и информационных направлений внедрения информационной технологии создает предпосылки для реализации новой интегрированной концепции применения информационных технологий в образовании. Сущность этой концепции заключается в реализации информационных технологий для личностно-ориентированного развития всех.

Сегодня совершенно ясно, что современный мир становится всё более зависимым от информационных технологий и будущее неизбежно потребует большого запаса разнообразных знаний, включая и знания информационных технологий.

Актуальность использования информационных технологий в современном образовании диктуется стремительным развитием информационного общества, широким распространением технологий мультимедиа, электронных информационных ресурсов, сетевых технологий, позволяющих использовать информационные технологии в качестве средства обучения.

В преподавании программирования основной задачей является обучение составлению алгоритма программы на выбранном языке программирования. Современные электронные технологии, являясь инструментом обучения программированию, позволяют расширить традиционный подход к обучению. В настоящее время прослеживается явная тенденция перехода обучения от

алгоритмических к объектно-ориентированным языкам с использованием визуальных систем программирования.

Сегодня несложно проследить новый технологический виток, в рамках которого наблюдается наплыв визуальных сред разработки. В большинстве случаев они фактически не требуют навыков программирования, за счет чего привлекают огромное число пользователей различного уровня подготовки.

Одной из разновидностей широкого спектра доступных программ является визуальная среда программирования Kodu Game Lab, которая сочетает в себе и программирование, и графику, и моделирование, и которая в доступной и игровой форме может помочь обучающимся освоить темы школьного курса информатики.

Чему может научить Kodu? Kodu знакомит с логикой программирования и способами решения проблем, обходясь без сложного синтаксиса. Kodu включает условия и последовательности и является объектно-ориентированным. Kodu развивает реальные навыки XXI века, побуждая пользователей глубоко анализировать проблему и структурировать свое решение — подход, применимый ко всем учебным предметам, деловым и личным отношениям.

Важнейший компонент проекта Kodu — пользовательский интерфейс программирования. Язык очень прост, в нем используются только значки. Программы состоят из страниц, которые подразделяются на правила, которые, в свою очередь, подразделяются на условия и действия. Язык Kodu специально предназначен для разработки игр и содержит специализированные примитивы (существительные, прилагательные и глаголы) языка, взятые из игровых сценариев. Программы выражаются в физических терминах; для управления поведением персонажей используются такие понятия как зрение, слух и время. В Kodu сложные понятия проектирования игр выражаются просто, непосредственно и интуитивно понятно.

Объект исследования: среды объектно-ориентированного программирования Kodu Game Lab и Scratch.

Целью выпускной квалификационной работы является рассмотрение визуальной среды программирования Kodu Game Lab для создания учебного квеста, а также поиск возможностей использования данного квеста в образовательном процессе.

Для достижения указанной цели поставлены следующие задачи:

1. Изучить теоретические понятия "квест", "учебный квест", "технология".
2. Описать технологию разработки учебного квеста.
3. Выполнить сравнительный анализ, исследовав функциональные возможности сред объектно-ориентированного программирования Kodu Game Lab и Scratch.
4. Разработать учебный квест в среде объектно-ориентированного программирования Kodu Game Lab с различными уровнями и различными типами игр.

Учебный квест – это и игра, и обучение одновременно. Данная технология разнообразит процесс обучения школьников, сделает его интересней и ярче. Участие в квесте способствует формированию у учащихся навыков самостоятельной работы.

Занимательная форма квеста в Kodu Game Lab, приближенная к игре, заинтересует не только детей, но и их родителей. Привлечение к учебной деятельности родителей станет одним из самых ценных моментов использования учебных квестов. Они смогут увидеть, что нужно знать и уметь ребенку на данном этапе обучения и как реально ребенок самостоятельно справится с заданиями.

Итак, при использовании технологии учебного квеста в визуальном конструкторе Kodu Game Lab центральной действующей фигурой является ученик, а учитель выступает в роли помощника. Использование технологии учебного квеста не только повышает познавательную активность участников

образовательного процесса, но и способствует формированию ответственности школьников за собственное обучение.

Глава 1. АНАЛИЗ KODU GAME LAB ДЛЯ РАЗРАБОТКИ УЧЕБНОГО КВЕСТА

1.1. Основные понятия квеста

Квест или приключенческая игра — это продолжительный целенаправленный поиск, который может быть связан с приключениями или игрой, как один из основных жанров компьютерных игр, представляющий собой интерактивную историю с главным героем, управляемым игроком. Важнейшими элементами игры в жанре квеста являются собственно повествование и исследование мира, главную роль в игровом процессе играют решение головоломок и задач, требующих от игрока умственных усилий. Стандартная игра построена по цепочке: разгадаешь одно задание – получишь следующее. И так, пока не дойдешь до финиша. Такие характерные для других жанров компьютерных игр элементы, как бои, экономическое планирование и задачи, требующие от игрока скорости реакции и быстрых ответных действий, в квестах сведены к минимуму или отсутствуют.

Учебный квест — это образовательная игра-путешествие. Квест предполагает общую игровую ситуацию (сюжет), маршрутный лист с картой, нанесёнными на ней точками и заданиями для выполнения на точках.

Структура учебного квеста обычно включает в себя следующие компоненты:

1. Введение (изложение темы, обоснование значимости проекта);
2. Формулировку задания;
3. Распределение ролей;
4. Описание процесса работы;
5. Руководство к действиям;
6. Оценку полученных результатов;

7. Заключение (суммирование познавательных навыков, акцентирование возможности их применения в других областях).

Текстовые приключения:

Первые приключенческие игры — были текстовыми и использовали парсеры, основанные на управляющих глаголах. С развитием компьютерной техники и графических систем текстовые игры были вытеснены более наглядными и удобными в управлении (но и намного более ограничивающими возможности игрока) графическими квестами. Позднее любители текстовых игр возродили жанр *interactive fiction*, создав виртуальные машины для работы с файлами различных игр. Примером подобной игры является браузерная игра *Tale* и ее аналог *Легенды древности*. В настоящее время текстовые квесты чаще всего встречаются в качестве мини-игр в составе выпускаемых крупных игр.

Графические квесты:

Первые графические квесты появились для 8-битных домашних компьютеров в начале 1980-х. Однако по-настоящему «графическими» они стали лишь в тот момент, когда произошёл отказ от текстового интерфейса и переход к так называемому «point-and-click» (то есть управлению с помощью указателя посредством стрелок клавиатуры, джойстика или мыши), появившемуся в 1985. Одними из популярных игр этого поджанра являются серии игр *Monkey Island* и *Space Quest*.

Квесты - головоломки:

В квестах - головоломках (англ. *puzzle adventure game*) во главу угла ставится решение каких-либо логических задач, загадок, например, в виде различных механизмов, доступных для обследования игроком; при этом число загадок очень велико, а повествование может быть схематичным или вовсе отсутствовать. Ярким представителем поджанра является игра *Myst* (1993), и более поздние квесты-головоломки, подражающие ей, описывались игровыми обозревателями как «*Myst*-образные» квесты. Один из подвидов поджанра квестов-головоломок является «выход из комнаты», в котором перед игроком

стоит задача вывести персонажа из запертого помещения, пользуясь какими-либо находящимися в комнате предметами.

Визуальные романы:

Визуальный роман — характерный для Японии подвид текстового квеста, в котором развертываемая в виде текста история сопровождается статичными, реже анимированными изображениями. Степень интерактивности в таких играх обычно низка, и от зрителя лишь изредка требуется сделать определенный выбор с помощью меню, в частности — выбрать вариант ответа в диалоге. Визуальные романы крайне популярны в Японии, занимая значительную часть рынка японских игр для персонального компьютера. Популярные визуальные романы часто становятся основой для экранизации в виде аниме или выпуска манги.

Жанры игр:

Приключенческая игра (Adventure) — игра, обладающая полноценным литературным сюжетом, и игрок в процессе игры сам раскрывает все перипетии этого сюжета.

Ролевая игра (RPG — англ. Role Playing Game) — игра, отличительной особенностью которой является наличие у персонажей определённых навыков и характеристик, которые можно обрести, а впоследствии развивать, выполняя какие-либо действия.

Компьютерный симулятор (Simulator) — игра, полностью имитирующая какую-либо область реальной жизни, например, имитация управления гоночным автомобилем или самолётом.

Головоломка (Puzzle) — игра, полностью или более чем наполовину состоящая из решения различных логических задач и головоломок.

Образовательная игра — игра, включающая в себя элементы обучающих программ, которые подаются через сам игровой процесс и, благодаря повышению интереса к ним в связи с необычным антуражем, впоследствии хорошо запоминаются.

Забавы — игры, в основном рассчитанные на детей, где психологическое впечатление от происходящего на экране гораздо важнее самого процесса игры — например, вид лопающихся пузырьков.

Казуальная игра — компьютерная игра, предназначенная для широкого круга пользователей. Сам термин «казуальная» происходит от лат. *casualis*, что означает «случайный». Таким образом, казуальная игра — это игра, в которую играют от случая к случаю, между делом, чаще всего — чтобы как-то «убить» время. Ввиду своего предназначения такая игра, как правило, обладает достаточно простыми правилами и не требует от пользователя хорошего владения компьютером. Многие подобные игры обладают также яркой привлекательной графикой и минимумом текста.

Технология разработки учебного квеста:

1. Выбор темы и постановка цели.
2. Выбор игрового сюжета, формы работы с заданиями.
3. Создание игрового мира и ландшафта.
4. Помещение персонажей в игровой мир.
5. Установка правил поведения игры.

В общем виде сюжет — это своего рода базовая схема игры, включающая последовательность происходящих в ней действий и совокупность существующих отношений персонажей.

Проектирование сюжета, естественно, потребует:

- придумать и создать эскиз на бумаге: как будет выглядеть виртуальный игровой мир (ландшафт и его объекты: вода, дороги, деревья, горы и т.д.), в котором будут развиваться события;
- придумать героев (персонажей, объекты), управление которыми происходит в процессе игры (развитие сюжета) или которые будут действовать самостоятельно.

Сюжет игры во многом будет определяться возможностями тех объектов, которыми будет управлять играющий.

В любой игре используют такие элементы как:

- Основной персонаж/персонажи (главные герои).
- Второстепенные персонажи.

1.2. Сравнительный анализ продуктов Scratch и Kodu Game Lab для разработки учебного квеста

Визуальная технология программирования является современным подходом к созданию программ. В данный момент существуют различные толкования термина «визуальные языки программирования». Визуальное программирование — программирование, предусматривающее создание приложений с помощью наглядных средств.

Визуализация процесса создания программ позволяет на первых этапах увидеть результат своего труда, процесс создания интерфейса проектов увлекает учащихся, дает возможность проявить творческие способности.

Учащийся показывает, что должно получиться в результате, а текст программы генерируется автоматически с помощью визуального прототипа.

Визуальный язык программирования делает процесс создания программ наглядным и увлекательным и берет на себя большую часть рутинной работы.

Вместе с этим, данная технология позволяет разрабатывать достаточно сложные и профессиональные приложения. Визуальное программирование обладает достоинством наглядного представления информации и гораздо лучше соответствует природе человеческого восприятия, чем методы традиционного программирования.

Визуальное программирование - способ создания программ путем манипулирования графическими объектами вместо написания кода в текстовом виде.

Визуальное программирование позволяет программировать, используя графические или символьные элементы, которыми можно манипулировать

интерактивным образом согласно некоторым правилам, причем пространственное графических объектов использовать в качестве элементов синтаксиса программы. Значительная часть визуальных языков программирования базируется на идее «фигур и линий», где фигуры (прямоугольники, овалы и др.) рассматриваются как субъекты и соединяются линиями (стрелками, дугами и др.), которые представляют собой отношения.

Создание компьютерных игр, моделирование, презентации, учебные пособия — вот неполный перечень того, где можно использовать *Scratch* и *Kodu Game Lab*.

Scratch - это визуальная объектно-ориентированная среда программирования для обучения учащихся основной и средней школы. В свою очередь *Kodu Game Lab* - визуальный язык программирования, предназначенный для создания трёхмерных игр и миров. Выделяются некоторые основания для сравнения:

1. Возраст, для которого предназначена среда.

Разработчики *Scratch* и *Kodu Game Lab* убеждают нас в том, что *Scratch* и *Kodu Game Lab* предназначен для учащихся от 8 и до 16 лет. В обеих средах можно работать и с учащимися более старшего возраста, увеличивая сложность проекта, согласно имеющимся знаниям по сопряжённым предметам. В любом случае, обе среды визуальны и интуитивно понятны, и, если в *Scratch* мы перетаскиваем и складываем кирпичики, то в *Kodu Game Lab* мы выбираем действия из определённых кругов.

2. Условия распространения.

Обе программы распространяются бесплатно.

3. Платформы, на которых работает программное обеспечение.

Scratch (его локальная версия) имеет здесь явное преимущество, поскольку хорошо зарекомендовал себя на различных версиях *Windows*, *MacOS* и *Linux*. На сайте программы можно получить дистрибутив для разных платформ. Разработчиком *Kodu Game Lab* является *Microsoft Research (MSR)* —

подразделение корпорации *Microsoft*, соответственно, данная среда безотказно будет работать на любых версиях *Windows*, а другие операционные системы не рассматриваются.

4. Объём дистрибутива и системные требования.

Дистрибутив *Scratch* 2.0 для *Windows* весит 47,6 Мб, а *Kodu Game Lab* версия 1.4.111.0 весит 247.7 Мб. Для корректной работы со *Scratch*, особенно если планируется загрузка и просмотр проектов в сети, нужно дополнительно загрузить *Java*, а для работы *Kodu Game Lab* требуется *DirectX* и *.NET Framework*, о чём идет предупреждение во время установки.

5. Tutorial (обучение).

Scratch постепенно обрастает подсказками по командным ящикам, благодаря чему начинающие пользователи могут без страха осуществить первые шаги в этой среде. *Kodu Game Lab* также содержит пошаговые уроки для пользователей в отдельном разделе "Уроки", поэтому элементарные действия в программе также осваиваются легко и непринуждённо. Однако следует отметить, что полноценных пособий (с поурочными планами, методическими рекомендациями) для работы с учащимися не имеется ни по *Scratch*, ни по *Kodu Game Lab*.

6. Использование онлайн редактора.

Scratch предполагает работу как локально, так и использование онлайн редактора. С *Kodu Game Lab* предполагается только локальная работа над проектами.

7. Сетевые сообщества.

Сообщество *Scratch* и сообщество *Kodu Game Lab* чем-то похожи друг на друга: пользователи, примеры проектов (миров), возможность оценки и комментирования, форумы. Кроме того, *Scratch*-проекты просматриваются прямо онлайн при наличии *Java*. Проекты *Kodu Game Lab* можно просмотреть, лишь скачав на свой компьютер и открыв при помощи данной среды.

8. Богатство языка.

Scratch содержит в себе множество алгоритмических конструкций. Инструкции для объектов *Kodu Game Lab* выстраиваются на основе условного оператора, могут содержать вложенную логику, завязанную на работу сенсоров и действий. Таким образом, принципы программирования несколько разнятся. Проекты в *Scratch* двумерны, а в *Kodu Game Lab* трёхмерны, потому они и называются мирами.

9. Формат файлов (закрытость/открытость системы).

Обе среды имеют собственный формат файлов: *.sb* (*.sb2* для версии 2.0) и *.kodu2*. Этим и обусловлена их закрытость. Однако, как уже было сказано выше, *Scratch*-истории можно просматривать онлайн. С другой стороны, игры, созданные в *Kodu Game Lab*, могут работать и редактироваться с контроллером *Xbox*.

Преимуществом среды *Kodu* является возможность визуализации написанного сценария, что позволяет установить соответствие между планируемыми действиями исполнителя и его реальным поведением.

Kodu знакомит с логикой программирования и способами решения проблем, обходясь без сложного синтаксиса. *Kodu* включает условия и последовательности и является объектно-ориентированным. *Kodu* развивает реальные навыки XXI века, побуждая пользователей глубоко анализировать проблему и структурировать свое решение — подход, применимый ко всем учебным предметам, деловым и личным отношениям.

Таким образом, трехмерный образ миров *Kodu Game Lab* более увлекателен и обширен. А значит будет интересен разновозрастным группам пользователей.

1.3 Техническое задание

Техническое задание на разработку учебного квеста в визуальной объектно-ориентированной среде KODU GAME LAB.

1. Общие сведения.

- 1.1. Муниципальное бюджетное общеобразовательное учреждение «Средняя общеобразовательная школа № 2» Асбестовского городского округа.
- 1.2. Действия с файлами в операционной системе Windows.
- 1.3. Обучение пользователей ПК.
- 1.4. Сентябрь 2015 – май 2016.

2. Характеристика области применения продукта.

- 2.1. Использование продукта разработки возможно в образовательном процессе.
- 2.2. Учителя информатики, учащиеся 5-9 классов.

3. Требования к продукту разработки.

- 3.1. Учебный квест должен содержать несколько различных уровней с различными типами игр.
- 3.2. Видеокарта, поддерживающая технологию DirectX9.0c и Shader Model 2.0 или выше.
- 3.3. Операционные системы: Windows 8 или 8.1(в режиме рабочего стола); Windows 7; Windows Vista; Windows XP (SP3).
- 3.4. .NET Framework 4.0 или выше и XNA Framework 4.0 (если на компьютере не установлено данное ПО, программа установки предложит установить эти продукты).
- 3.5. Управление происходит с помощью клавиатуры и мыши. Продукт сохраняется в собственном расширении «.Kodu2».

- 3.6. Ввод данных происходит с помощью клавиатуры и мышки, либо с помощью геймпада Xbox360. Данные выводятся на монитор. Возможен экспорт игр в расширении «.Kodu2»
- 4. Требования к пользовательскому интерфейсу.
 - 4.1. Пользовательский интерфейс отсутствует. Доступ к панели управления можно получить с помощью нажатия клавиши «Esc».
 - 4.2. Дизайн экрана минималистичен. Камера следует за Kodu и управляется с помощью мыши.
 - 4.3. Управление движением Kodu происходит с помощью клавиш W,a,s,d. Стрельба пулями с помощью клавиши «LeftShift». Стрельба ракетами с помощью клавиши «ПРОБЕЛ». Поднять предметы можно нажав клавишу «PageUp». Отпустить предмет клавишей «PageDown».
- 5. Требования к документированию.
 - 5.1. Отсутствуют.
- 6. Порядок сдачи-приемки продукта.

Глава 2. РАЗРАБОТКА УЧЕБНОГО КВЕСТА В ВИЗУАЛЬНОЙ СРЕДЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ KODU GAME LAB

2.1 Знакомство с визуальной объектно-ориентированной средой Kodu Game Lab

Общая характеристика программного продукта Kodu Game Lab.

Kodu Game Lab (англ. Лаборатория игр Коду) – это визуальная среда и язык программирования, предназначенные для создания трехмерных игр для PC и игровой консоли XBox 360.

Drag-and-drop программирование используется с так называемым языком высокого уровня Kodu Language. С его помощью можно управлять параметрами и настройками игрового мира, коллизиями, цветом и визуализацией.

Графический интерфейс обладает высокой степенью прозрачности, за счет чего пользователь легко манипулирует исполнителями на экране, создает поведение игрока, управляет визуальным рядом, звуками и сценарием.

Для создания миров используется библиотека, включающая более 200 стандартных игровых сценариев и базовых игровых элементов, редактор игрового поведения, 20 различных персонажей с различными способностями.

Программа в Kodu - это набор правил, которые определяют действия объекта. Для написания правил в Kodu используются два оператора:

When<условие>Do<действие>

When (англ. “когда”, “если”, “в то время как”) - оператор, определяющий условие;

Do (англ. “делать”) - оператор, определяющий непосредственное действие, которое должен выполнить объект при соответствующем условии. Программа создается для каждого объекта индивидуально.

Изначально среда Kodu разрабатывалась таким образом, чтобы пользователи играли в игры и редактировали миры с помощью контроллера Xbox. Но поскольку у многих общественных центров, учебных заведений и домашних пользователей может не хватить средств на покупку контроллеров, разработчики Kodu создали версию среды, рассчитанную на работу с клавиатурой. Однако в большинство игровых миров Kodu еще не внесены соответствующие изменения, и для их использования в играх необходимы контроллеры. Это может вызывать проблемы у пользователей, которые недавно начали осваивать среду, но в то же время стимулирует усовершенствование пользователями существующих миров путем редактирования кода.

Существуют два основных метода, позволяющих управлять аватаром или персонажами с помощью клавиатуры.

При использовании первого метода клавишам курсора сопоставляются общие направления. Для этого достаточно поместить в код конструкцию `WHEN: keyboard DO: Move`. При этом клавиша со стрелкой вверх соответствует движению вперед, со стрелкой вниз - движению назад, а клавиши со стрелками влево и вправо — поворотам влево и вправо соответственно. Направления сторон света (север, юг, запад и восток) при использовании этого метода не учитываются.

При использовании второго метода клавишами курсора сопоставляется движение в конкретном направлении (север — вверх, юг — вниз, запад — вправо, восток — влево). Однако этот метод может приводить к путанице, поскольку в зависимости от особенностей ландшафта и местоположения персонажей направления на север и вверх, на юг и вниз и т. д. могут различаться. Несмотря на это, игровые миры и персонажей можно изменить таким образом, чтобы управлять игрой с помощью клавиатуры и мыши. При использовании данного метода для перемещения по игровому миру с помощью стрелок курсора необходимо учитывать показания компаса. В интерфейсе некоторых миров компас отсутствует. Чтобы отобразить его, перейдите в

режим редактирования, нажав клавишу Esc, выберите последний значок в списке (значок с изображением гаечного ключа и ландшафта с горой), стрелкой вниз прокрутите список до пункта Show Compass («Показывать компас») и активируйте этот элемент.

Инструкция по установке Kodu Game Lab.

Скачать среду программирования можно, перейдя по ссылке: <https://www.microsoft.com/ru-ru/softmicrosoft/kodu.aspx>

Перед установкой программы ОЗНАКОМИТЬСЯ с требованиями к ПК:

1. ПК должен работать под управлением ОС Windows.
2. ПК должен поддерживать графику DirectX9.
3. ПК должен поддерживать Shader Model версии 2.0 или более поздней.

Определение версии DirectX:

- Для Windows 7 и Windows Vista: в меню «Пуск» введите в поле поиска строку Cmd. Для Windows XP: в меню «Пуск» выберите пункт «Выполнить» и введите команду Cmd.
- Выполните в открывшемся окне команду DxDiag. Версия DirectX будет указана на первой вкладке.

Если на компьютере не установлено ПО .NET Framework версии 3.5 и XNA Game Studio версии 3.1, программа установки предложит установить эти продукты.

Оптимизация Kodu для работы на ПК:

Если изображение в игре движется плавно, значит видеоадаптер обеспечивает достаточную частоту смены кадров. Для комфортной игры необходимо, чтобы частота смены кадров составляла не менее 20 кадров в секунду.

Если при получении команд Kodu движется рывками, воспользуйтесь приведенными ниже рекомендациями, чтобы повысить удобство игры.

Определение частоты смены кадров:

Запустите приложение Configure Kodu Game Lab из меню программ на ПК. Появится панель инструментов настройки Kodu. Установите флажок Show Frames per Second («Показывать число кадров в секунду»).

При следующем запуске Game Lab на экране будет отображаться количество кадров в секунду (Frames per Second, fps). Отображение числа кадров в секунду можно включать и отключать в любой момент, устанавливая или снимая флажок Show Frames per Second в панели инструментов настройки.

Повышение частоты смены кадров:

Если в панели инструментов настройки выбран параметр Shader Model 3 («Модель построителя текстуры версии 3»), выберите вместо него параметр Shader Model 2 («Модель построителя текстуры версии 2»). Снимите флажок Fullscreen («Полноэкранный режим»). Выберите в раскрывающемся списке Resolution («Разрешение») меньшее разрешение. При этом уменьшится детализация изображения, и приложение будет потреблять меньше ресурсов процессора. Снимите три флажка визуальных эффектов: glow («Свечение»), smoothing («Сглаживание») и animation («Анимация»).

Примечание для пользователей ноутбуков:

Некоторые ноутбуки со встроенными видеоадаптерами не поддерживают ряд возможностей Kodu, но позволяют использовать основные средства Kodu : играть и создавать учебные игры.

Три основных этапа разработки в среде Kodu Game Lab:

1. Создаем (рисуем) игровой мир.



Рисунок 2.1. Пример игрового ландшафта

2. Помещаем персонажей в игровой мир.
3. Устанавливаем (программируем) правила поведения персонажей и объектов.

Для создания игрового мира необходимо:

1. Запустить программу



Рисунок 2.2. Ярлык Kodu Game Lab

2. Выбрать в главном меню «New world» или «Новый пустой мир»



Рисунок 2.3. Главное меню Kodu Game Lab

3. Откроется окно редактирования и панель инструментов

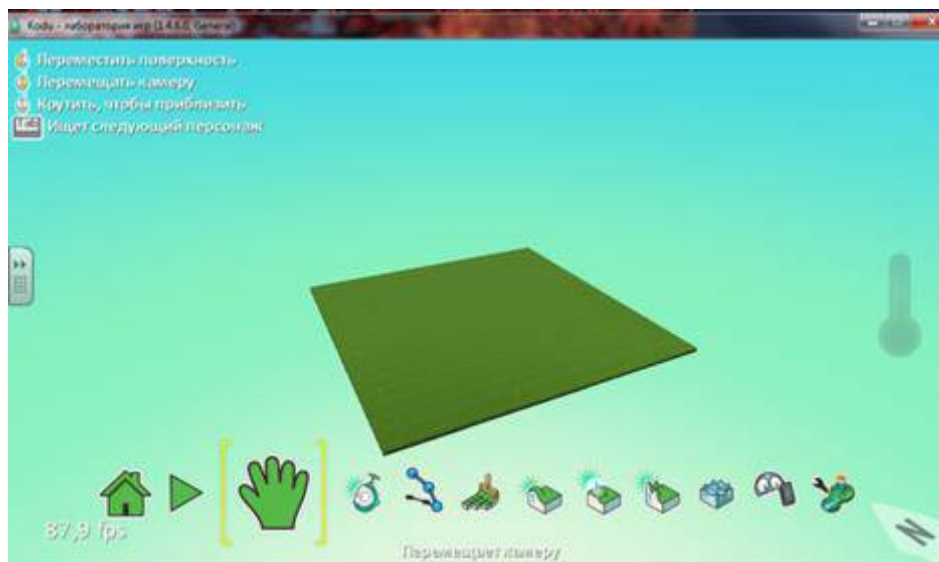


Рисунок 2.4 Игровое поле Kodu Game Lab

Для управления игровым миром:
Выберите инструмент «Рука» или удерживайте клавишу «Пробел»



Рисунок 2.5 Инструмент «Рука»

С помощью колесика мыши можно приближать и удалять мир. Правая кнопка мыши нужна для поворота, вращения и наклона камеры в трехмерном пространстве. Удерживая левую кнопку мыши, пользователь сможет переместить мир по горизонтальной поверхности.

Для создания ландшафта:
Выберите инструмент «Кисть для земли»



Рисунок 2.6 Инструмент «Кисть для земли»

Удерживая левую кнопку мыши, создается новый участок земли. С помощью правой кнопки мыши, можно удалить участок. Размер кисти задается с помощью клавиш управления курсором на клавиатуре. Доступен большой выбор текстуры ландшафта



Рисунок 2.7 Пример доступных текстур ландшафта

Среда Kodu Game Lab позволяет создать холмистую поверхность с помощью инструмента «Неровности», и заполнить их водой с помощью инструмента «Вода»



Рисунок 2.8 Инструмент «Неровности»



Рисунок 2.9 Инструмент «Вода»

Чтобы создать территорию, ее нужно нарисовать с помощью средств формирования ландшафта. Доступен выбор цвета и типа кисти (квадрат, овал, квадратная линия, овальная линия). Чтобы быстро создать территорию, достаточно увеличить размер кисти, нажимая клавишу со стрелкой вправо, и

«нарисовать» один квадрат. Остальные средства формирования ландшафта позволяют создавать возвышенности и впадины и делать их более округлыми или более резкими. Все водные поверхности должны создаваться только поверх суши. По умолчанию каждый мир имеет «стеклянные стены», которые удерживают воду и не дают предметам падать за края. После формирования местности к миру можно добавлять и программировать объекты и персонажей, как описано выше. Используя простые инструменты можно создавать детальные трехмерные ландшафты, управление освещением и камерой.

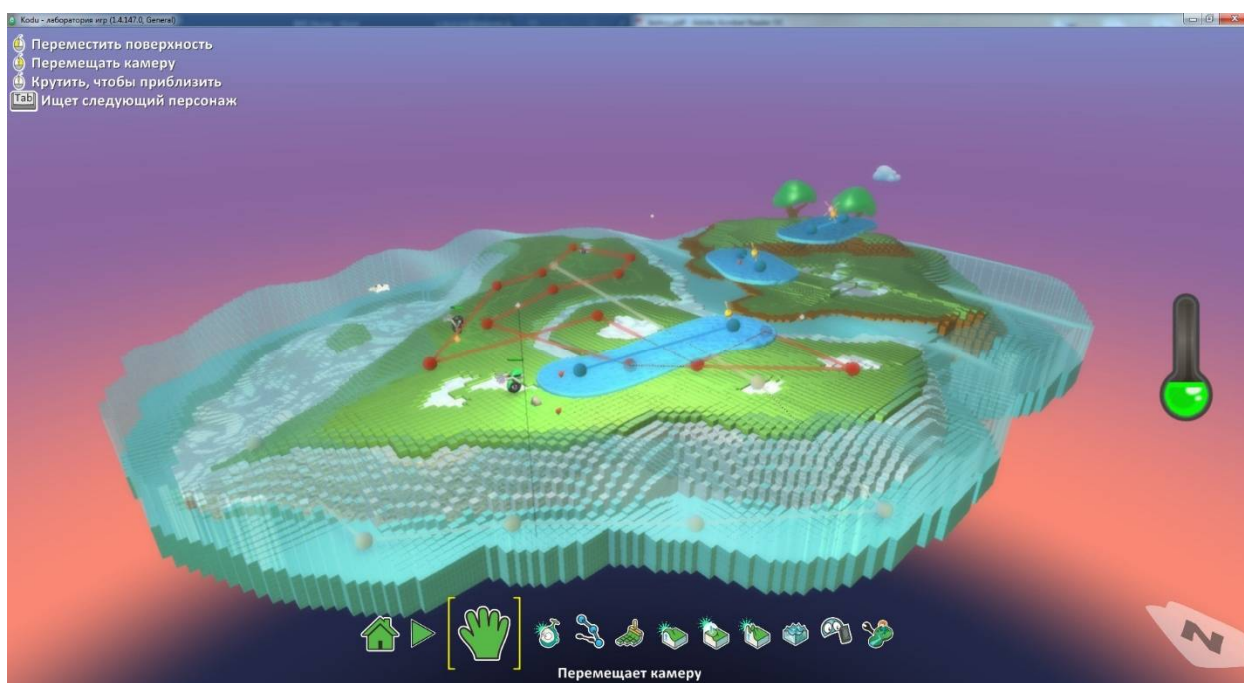


Рисунок 2.10 Пример ландшафта

Для того чтобы поместить персонажей в игровой мир:

Выберите инструмент «Объект», нажмите левой кнопкой мыши по игровой площадке и вставьте выбранный объект на игровое поле.



Рисунок 2.11 Инструмент «Объект»



Рисунок 2.12 Доступные персонажи

Предусмотрено 20 типов персонажей, обладающих различными характеристиками: kodu, rover, шар, байкер, летающая рыба, самолет, octopus, тарелка, подлодка, пушка, аэростат, батут, корабль, мина, шайба, тумба, спутник, задвижка, рыба, черепаха. Значок Kodu позволяет добавлять новые объекты и персонажей, щелкнув в пустом месте, или редактировать существующий объект.

Возможности персонажа Kodu:

1. Kodu может смотреть, слышать и различать цвета.
2. Kodu может выражать любовь, гнев, грусть и бурные эмоции.
3. Kodu может говорить, отображая текстовые сообщения.
4. Kodu может ожидать в определенных местоположениях.
5. Kodu может перемещаться между местоположениями за определенное время.
6. Персонаж Kodu может двигаться по направлению к определенным объектам и удаляться от них.

Чтобы массово удалить ненужные объекты, используйте инструмент «Удалить»



Рисунок 2.13 Инструмент «Удалить»

Для установки правил поведение персонажей и объектов:

Кликните по объекту правой кнопкой мыши. Выберите в контекстном меню пункт «Программа».

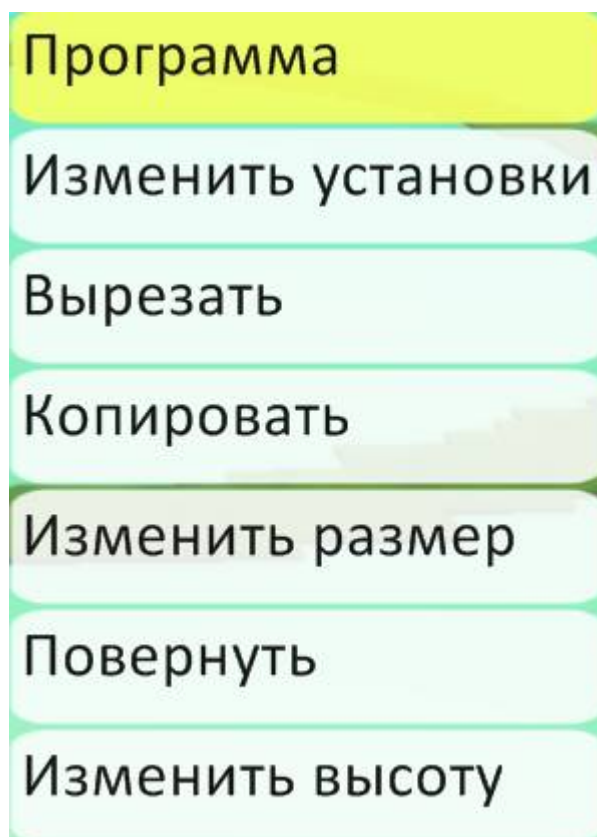


Рисунок 2.14 Контекстное меню

Напишите программу, используя визуальные блоки.



Рисунок 2.15 Пример программы

Запуск режима игры происходит нажатием левой клавиши мыши на кнопку «Играть», либо клавишей «Esc» на клавиатуре (она же возвращает в режим редактирования).



Рисунок 2.16 Кнопка «Играть»

Чтобы сохранить игру на переносные устройства памяти или персональный компьютер, в главном меню найдите пункт «Загрузить мир», где отображаются все игры, и выберите свою игру. Появится меню с элементами Play (Играть), Export (Экспорт) и Delete (Удалить). Выберите вариант Export (Экспорт). Игра будет сохранена в папку My Documents/SavedGames/Boku/Player1/Export... в формате .Kodu2, который удобно отправлять по электронной почте или сохранять на переносных устройствах памяти.



Рисунок 2.17 Сохранение игры

2.2 Создание учебного квеста в визуальной объектно-ориентированной среде Kodu Game Lab

Рассмотрев технические возможности визуального конструктора Kodu Game Lab, можно определить темы будущих уровней в учебном квесте:

- 1) Форматирование быстрое и полное.
- 2) Дефрагментация.
- 3) Компьютерный вирус.
- 4) Компьютерный антивирус.

Для создания уровней учебного квеста необходимо углубиться в содержание выбранных тем.

Уровень 1:

Форматирование – это процесс, во время которого происходит разметка носителя информации, а именно области хранения данных носителя. Существуют несколько способов этого процесса. Высокоуровневое форматирование делят на два вида — быстрое и полное. При быстром форматировании заново создается лишь файловая система, при этом область данных не затрагивается. При полном форматировании вначале производится полная проверка поверхности жесткого диска на наличие поврежденных секторов. Если такие сектора обнаруживаются, то программа проверки пытается их исправить, а при неудаче помечает их как неисправные. Соответственно, такие сектора исключаются из таблиц файловой системы, которые формируются после проверки поверхности. Это означает, что в дальнейшем в неисправные сектора данные просто не записываются. В отличие от быстрого форматирования, полное форматирование включает в себя еще и предварительную проверку целостности жесткого диска, то есть поиск поврежденных секторов. Так как при этой операции проверяется каждый сектор диска отдельно, то полное форматирование занимает на порядок больше

времени, чем быстрое. Время полного форматирования зависит от величины форматируемого раздела.

Зная процесс форматирования информации, был разработан сюжет и задания для первого уровня в учебном квесте. Главным героем в уровне «Форматирование быстрое и полное» стал персонаж Kodu, который, используя свое оружие, должен уничтожить все файлы – монеты. Перед тем как начать игру, обучающийся увидит описание уровня «Форматирование быстрое и полное». Ознакомившись с понятиями быстрого и полного форматирования, обучающийся сможет приступить к прохождению первого уровня учебного квеста в игровой форме. Используя оружие персонажа Kodu: пульки и ракеты, обучающийся должен уничтожить все монеты (файлы). Только после того, как игрок выполнит условия данного уровня, он получит доступ к следующему этапу учебного квеста.

- 1 Этап разработки – создание игрового ландшафта;
- 2 Этап разработки – создание персонажей и объектов;
- 3 Этап разработки – написание алгоритма для персонажа и объектов.

Уровень 2

Дефрагментация — процесс обновления и оптимизации логической структуры раздела диска для обеспечения хранения файлов в непрерывной последовательности кластеров. После дефрагментации ускоряется чтение и запись файлов, а, значит, и работа программ, ввиду того, что последовательные операции чтения и записи выполняются быстрее случайных обращений (например, для жесткого диска при этом не требуется перемещение головки). Другими словами – это перераспределение файлов на диске, при котором они располагаются в непрерывных областях.

Длинные файлы занимают сразу несколько кластеров. Если запись производится на незаполненный диск, то кластеры, принадлежащие одному файлу, записываются поочередно. Если диск переполнен, на нём может не быть

цельной области, достаточной для размещения конкретного файла. Тем не менее, этот файл все-таки запишется, если на диске много мелких областей, суммарный размер которых достаточен для записи. В таком случае файл записывается в виде нескольких фрагментов. Такое разбиение файла на фрагменты для записи на диск называется фрагментацией. Если на диске много фрагментированных файлов, скорость их чтения с носителя уменьшается, так как перемещение между кластерами, в которых хранятся фрагменты, на жёстких дисках требует времени. При использовании флеш-памяти время поиска значительно ниже, поэтому для них дефрагментация не требуется.

Для быстрого действия компьютера пользователь должен периодически производить дефрагментацию жесткого диска, поэтому уместно будет ознакомить обучающихся с данной процедурой. Так возник сюжет 2 уровня учебного квеста в визуальной среде программирования Kodu Game Lab. Главным героем второго уровня остался персонаж Kodu. Изменился способ прохождения уровня: необходимо перенести все яблоки (файлы) в сектора кластерной сетки, соответствующие цвету яблока (файла). Верно выполнив требования соответствий, обучающийся получает доступ к следующему уровню.

1 этап разработки – создание ландшафта:

Создана прямоугольная поверхность игрового уровня, для дизайна ландшафта использовался материал под номером 115. Ячейки кластерной сетки закрашены материалами ландшафта 42, 45, 118 и 119.

2 этап разработки – создание персонажей и объектов:

Создан главный персонаж Kodu, а также 24 яблока (файла) разного цвета, символизирующие ячейки кластерной сетки. Кластерная сетка создана из объектов «straight pipe», «cross pipe» и «corner pipe».

3 этап разработки – написание алгоритма для персонажа и объектов:

Таблица 2.1.

Алгоритм Kodu

Когда	Делать
Зажаты клавиши WASD	Двигаться
Нажата клавиша PageUp	Взять предмет
Нажата клавиша PageDown	Уронить предмет
Когда белый счет равен 24 очкам	Переход на следующий уровень
Kodu видит яблоко	Сказать: «Переместите файлы (яблоки) в кластерную сетку (в квадратные поля) с помощью Kodu. Поднять яблоко можно с помощью клавиши PageUp, отпустить – PageDown. Для каждого ряда в кластерной сетке используйте одинаковые файлы по типу (по цвету).» один раз. Данное действие предусмотрено как инструкция для пользователя вначале уровня.

Таблица 2.2.

Алгоритм яблок

Когда	Делать
Объект соприкасается с определенным типом ландшафта.	В белый счет добавить одно очко один раз.

При соприкосновении яблока определенного цвета с ячейкой в кластерной сетке этого же цвета добавляется одно очко в белый счет.

Уровень 3

Скачанные файлы на ПК из глобальной сети интернет бывают заражены различными вирусами. Нередко сами разработчики своих программ умышленно внедряют в свои продукты вредоносное программное обеспечение. И часто даже для уверенных пользователей это вредоносное программное обеспечение может нанести огромный вред их персональному компьютеру.

Компьютерный вирус — вид вредоносного программного обеспечения, способного создавать копии самого себя и внедряться в код других программ, системные области памяти, загрузочные секторы, а также распространять свои копии по разнообразным каналам связи.

Как правило, целью вируса является нарушение работы программно-аппаратных комплексов: удаление файлов, приведение в негодность структур размещения данных, блокирование работы пользователей или же приведение в негодность аппаратных комплексов компьютера и т. п. Даже если автор вируса не запрограммировал вредоносных эффектов, вирус может приводить к сбоям компьютера из-за ошибок, неучтённых тонкостей взаимодействия с операционной системой и другими программами. Кроме того, вирусы, как правило, занимают место на накопителях информации и потребляют некоторые другие ресурсы системы.

В обиходе «вирусами» называют всё вредоносное программное обеспечение, хотя на самом деле это лишь один его вид.

Классификация

Ныне существует немало разновидностей вирусов, различающихся по основному способу распространения и функциональности. Если изначально вирусы распространялись на дискетах и других носителях, то сейчас доминируют вирусы, распространяющиеся через Интернет. Растёт и функциональность вирусов, которую они перенимают от других видов программ.

В настоящее время не существует единой системы классификации и именования вирусов (хотя попытка создать стандарт была предпринята на встрече CARO в 1991 году). Принято разделять вирусы:

- по поражаемым объектам (файловые вирусы, загрузочные вирусы, сценарные вирусы, макровирусы, вирусы, поражающие исходный код);
- файловые вирусы делят по механизму заражения: паразитирующие добавляют себя в исполняемый файл, перезаписывающие невосстановимо портят заражённый файл, «спутники» идут отдельным файлом.
- по поражаемым операционным системам и платформам (DOS, Windows, Unix, Linux, Android);
- по технологиям, используемым вирусом (полиморфные вирусы, стелс-вирусы, руткиты);
- по языку, на котором написан вирус (ассемблер, высокоуровневый язык программирования, сценарный язык и др.);
- по дополнительной вредоносной функциональности (бэкдоры, кейлоггеры, шпионы, ботнеты и др.).

На 3 уровне учебного квеста обучающиеся знакомятся с понятием компьютерный вирус. Игрок, используя оружие персонажа Kodu, должен уничтожить все вирусы, которые представлены в форме рыбок. Они свободно перемещаются между файлами (монетами) и с периодичностью в 10 секунд уничтожают файлы. Главной задачей этого уровня является уничтожение всех вирусов до того, как они уничтожат все файлы

1 этап разработки – создание ландшафта

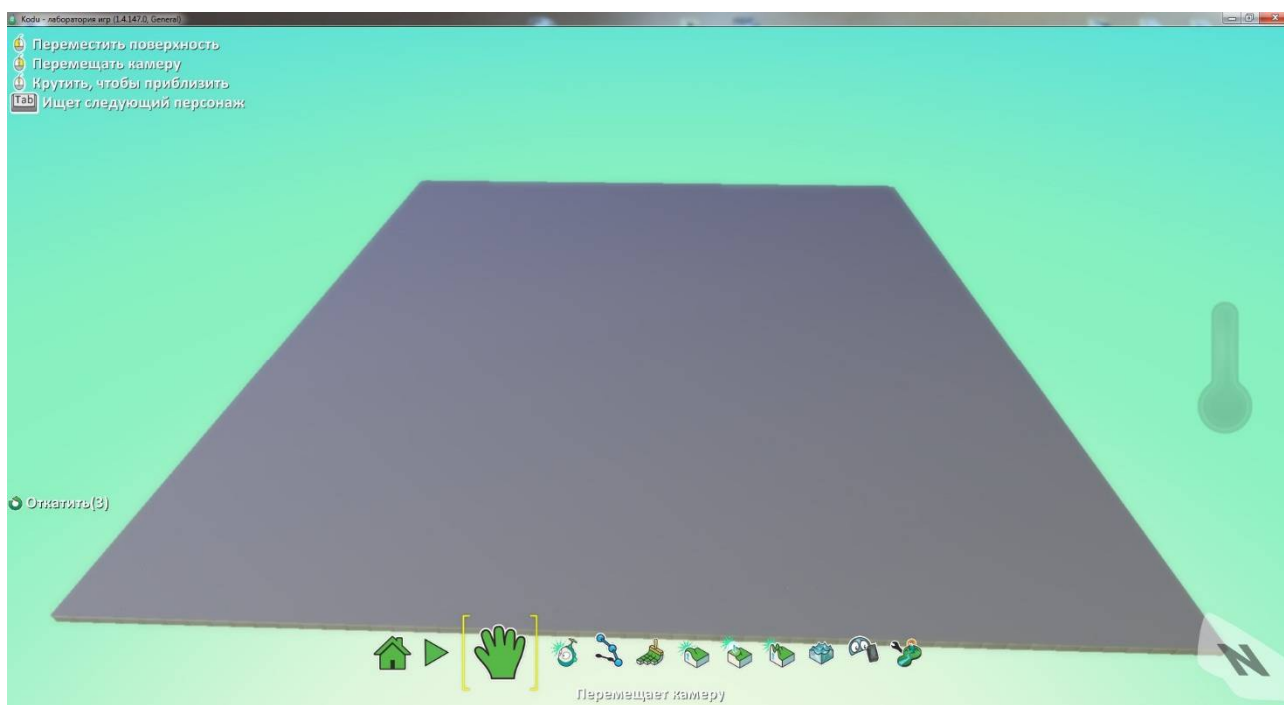


Рисунок 2.18 Ландшафт 3 уровня

2 этап разработки – создание пути движения вирусов

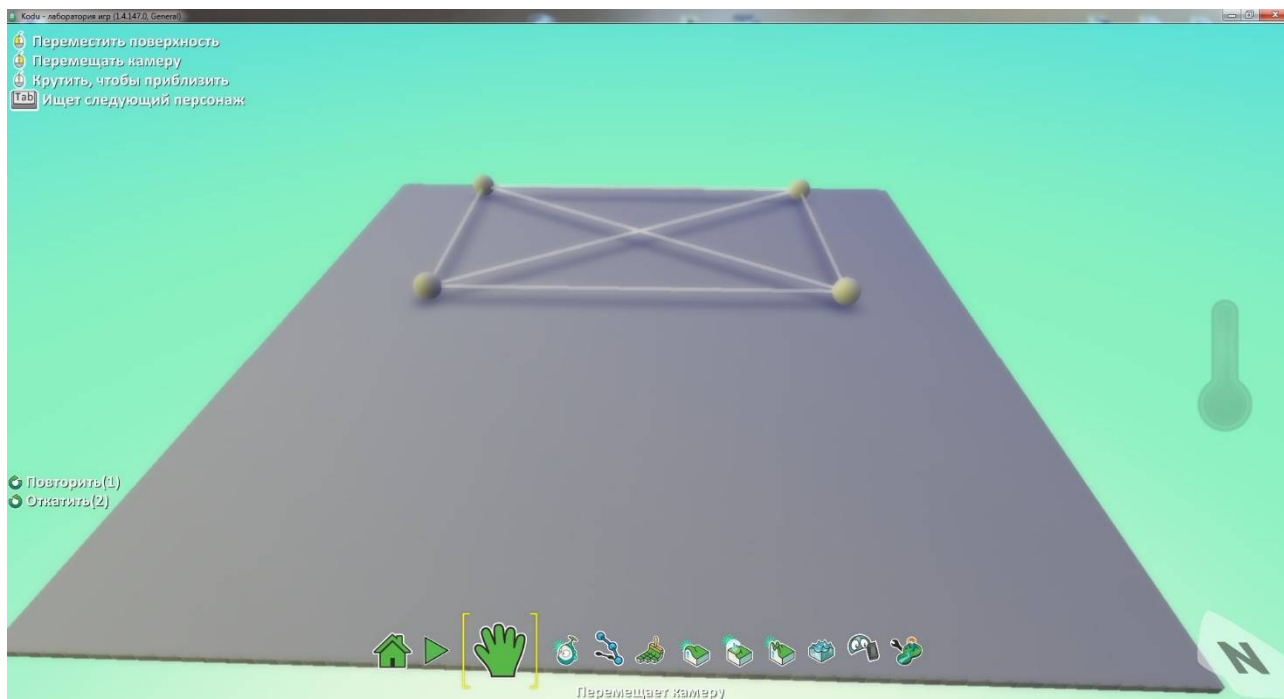


Рисунок 2.19 Пути движения

3 этап разработки – создание персонажей и объектов



Рисунок 2.20 Созданы персонажи и объекты

4 этап разработки – написание алгоритма для персонажей и объектов

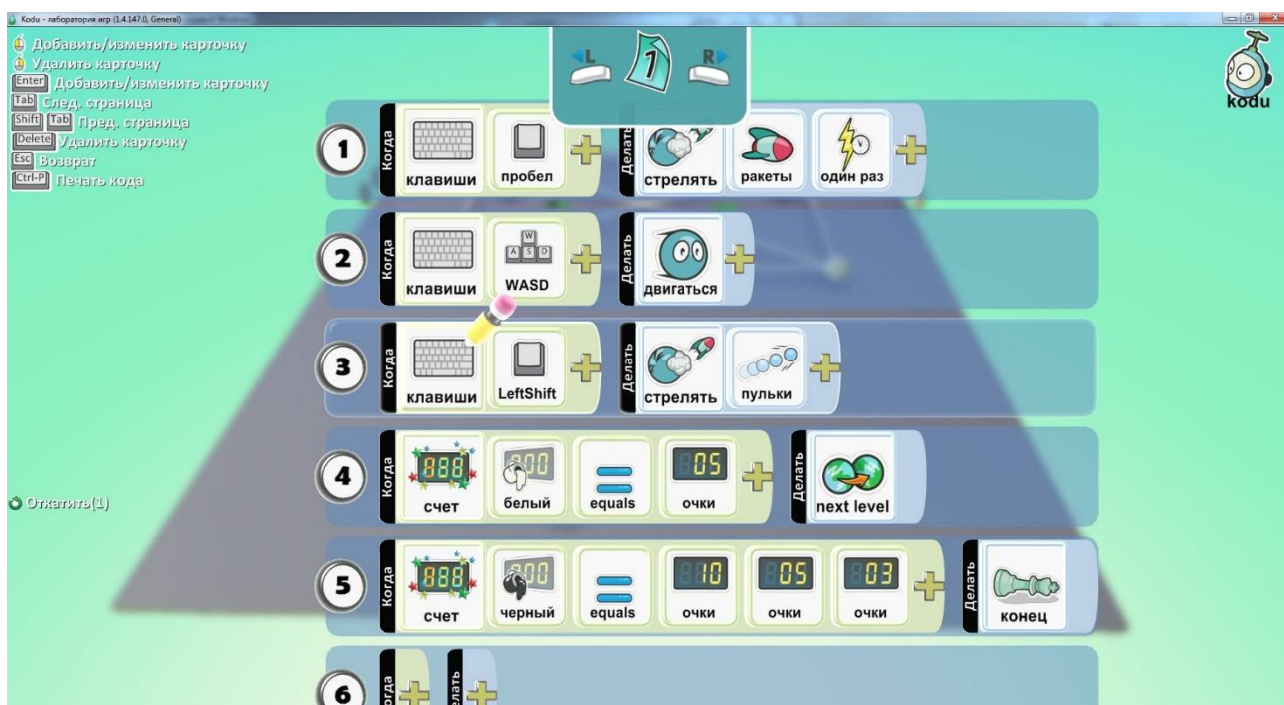


Рисунок 2.21 Алгоритм персонажа «Kodu»



Рисунок 2.22 Алгоритм персонажей «Летающая рыба» страница 1



Рисунок 2.23 Алгоритм персонажей «Летающая рыба» страница 2

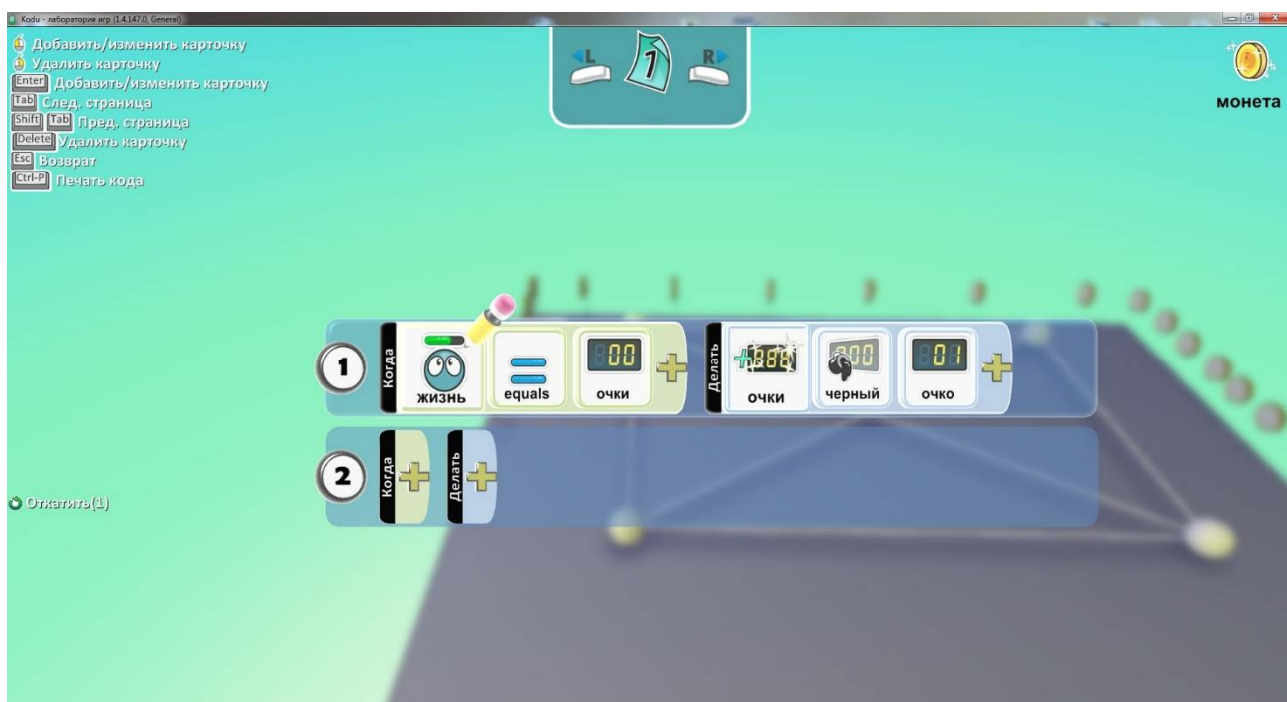


Рисунок 2.22 Алгоритм предметов «Монета»

Антивирусная программа (антивирус) — это такая специализированная программа для обнаружения компьютерных вирусов, а также нежелательных (считающихся вредоносными) программ вообще и восстановления заражённых (модифицированных) такими программами файлов, а также для профилактики — предотвращения заражения (модификации) файлов или операционной системы вредоносным кодом.

Классификация антивирусных продуктов:

1. По используемым технологиям антивирусной защиты:

- Классические антивирусные продукты (продукты, применяющие только сигнатурный метод детектирования, продукты, применяющие только проактивные технологии антивирусной защиты);
- Комбинированные продукты (продукты, применяющие как сигнатурные методы защиты, так и проактивные)

2. По функционалу продуктов:

- Антивирусные продукты (продукты, обеспечивающие только антивирусную защиту)
- Комбинированные продукты (продукты, обеспечивающие не только защиту от вредоносных программ, но и фильтрацию спама, шифрование и резервное копирование данных и другие функции)

3. По целевым платформам:

- Антивирусные продукты для ОС семейства Windows
- Антивирусные продукты для ОС семейства *NIX (к данному семейству относятся ОС BSD, Linux и др.)
- Антивирусные продукты для ОС семейства MacOS
- Антивирусные продукты для мобильных платформ (Windows Mobile, Symbian, iOS, BlackBerry, Android, Windows Phone 7 и др.)

4. Антивирусные продукты для корпоративных пользователей можно также классифицировать по объектам защиты:

- Антивирусные продукты для защиты рабочих станций
- Антивирусные продукты для защиты файловых и терминальных серверов
- Антивирусные продукты для защиты почтовых и Интернет-шлюзов
- Антивирусные продукты для защиты серверов виртуализации

После изучения понятия Компьютерный вирус необходимо познакомить обучающегося с возможностью уничтожения вирусов с помощью компьютерных антивирусов.

1 этап разработки – добавление нового персонажа в уровень «Компьютерный вирус».

2 этап разработки – написание алгоритма поведения для персонажа «Самолет 1»:

2.3 Акт внедрения учебного квеста в образовательный процесс

Российская Федерация
 Министерство общего и профессионального
 образования
 Свердловской области
 Управление образованием
 Асбестовского городского округа
**Муниципальное бюджетное
 общеобразовательное учреждение
 «Средняя общеобразовательная школа № 2»
 Асбестовского городского округа
 (средняя школа № 2)**
 улица Садовая, 13, город Асбест,
 Свердловская область, 624272, Россия
 телефон/факс 8 (34365) 7-69-08
 E-mail: scool_2asbest@mail.ru
 ОКПО 50310349
 ОГРН 1026600633938
 ИНН 6603010955 КПП 660301001

____ 18.11.2016 г. ____ № ____

На № ____ от _____

Настоящий акт подтверждает внедрение результатов выпускной квалификационной работы студента Уральского государственного педагогического университета Ивлева Павла Алексеевича «Разработка учебного квеста в визуальной среде объектно-ориентированного программирования Kodu Game Lab».

Основные результаты выпускной квалификационной работы: разработан учебный квест в среде объектно-ориентированного программирования Kodu Game Lab по теме: «Действия с файлами в операционной системе Windows».

Результаты работы были внедрены в учебный процесс Муниципального бюджетного общеобразовательного учреждения «Средняя общеобразовательная школа № 2» Асбестовского городского округа в 2016-2017 учебном году.

Учебный квест по теме «Действия с файлами в операционной системе Windows» прошел апробацию на уроках информатики в 8-х классах. Показателем результативности внедрения учебного квеста в образовательный процесс стали положительные результаты учащихся по предмету.

В результате достигнута положительная динамика по формальным и неформальным показателям качества образования учащихся. Изменился образовательный процесс с точки зрения использования современных информационных технологий, методов и приемов обучения учащихся. Представленный продукт полностью соответствует техническому заданию на его разработку.

Директор школы № 2

В.Ф.Кокшаров

М.П.

Учитель информатики

Е.В.Блохина

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была достигнута поставленная цель и выполнены все задачи, которые были сформулированы во введении.

В 1 главе выпускной квалификационной работы рассмотрены и изучены понятия «квест», «учебный квест» и технология разработки учебного квеста, описаны основные этапы разработки учебного квеста и его структура. Во-вторых, исследованы функциональные возможности сред объектно-ориентированного программирования Kodu Game Lab и Scratch и произведен их сравнительный анализ. Выявлены положительные и отрицательные функции каждой среды объектно-ориентированного программирования. Отмечены преимущества визуального конструктора Kodu Game Lab для создания учебного квеста. В-третьих, составлено техническое задание на разработку учебного квеста в визуальной объектно-ориентированной среде Kodu Game Lab.

Во 2 главе выпускной квалификационной работы дана общая характеристика программного продукта Kodu Game Lab и краткое руководство пользователя, которое дополнено инструкцией по установке Kodu Game Lab. Во-вторых, сформулированы темы учебного квеста, которые вошли в собственный продукт, разработанный в визуальной объектно-ориентированной среде Kodu Game Lab. Дан теоретический материал по темам: «Форматирование быстрое и полное», «Дефрагментация», «Компьютерный вирус», «Компьютерный Антивирус».

Внедрение визуальных конструкторов во все сферы программирования неизбежно. Последствия такого нашествия могут оказаться для многих сфер даже полезными. Результаты работы были внедрены в учебный процесс Муниципального бюджетного общеобразовательного учреждения «Средняя общеобразовательная школа № 2» Асбестовского городского округа в 2016-

2017 учебном году. Учебный квест по теме «Действия с файлами в операционной системе Windows» прошел апробацию на уроках информатики в 8-х классах. Показателем результативности внедрения учебного квеста в образовательный процесс стали положительные результаты учащихся по предмету.

Список информационных источников

1. Жемчужников, Д.Г. Обучение программированию на основе создания динамических игр: учебно-методическое пособие / Д.Г. Жемчужников. – Москва, Мэйлер, 2010. – 60 с.
2. Арсак Ж. Программирование игр и головоломок. Наука. Гл. ред. физ.-мат. лит., 1990.
3. Бадд Т. Объектно-ориентированное программирование в действии. - СПб.: "Питер", 1997.
4. Захаров А.А., Захарова И.Г. Языки и среды программирования. Учебное пособие. Тюмень: Изд. ТюмГУ, 2006. – 288 с.
5. Мухортов В.В., Рылов В.Ю. Объектно-ориентированное программирование, анализ и дизайн. Новосибирск: ИМСО РАН, 2002.
6. Попов, А.И., Использование веб-квестов в процессе организации профессиональной творческой подготовки студентов по приоритетным направлениям / А.И. Попов, В.Г. Однолько, А.А. Букин // Вопросы современной науки и практики. Университет им. В.И. Вернадского. – 2013. - № 4 (48). – С. 64-70.
7. Карденахлишвили, Т.Д. Компьютерные игры как технология коммуникативной виртуальной реальности – сети Интернет/Т.Д. Карденахлишвили // Каспийский регион: политика, экономика, культура. – 2012. – №1. – С. 316-322.
8. Антивирусная программа Википедия URL: https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D1%82%D0%B8%D0%B2%D0%B8%D1%80%D1%83%D1%81%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0 (дата обращения: 17.10.2016)
9. Компьютерный вирус // Википедия URL: <https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8>

- C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D0%B2%D0%B8%D1%80%D1%83%D1%81 (дата обращения: 17.10.2016).
- 10.Scratch URL: <https://scratch.mit.edu> (дата обращения: 14.10.2016)
- 11.Разновидности квестов URL: <http://mylsa.ru/xo15pliovl93/%D0%9A%D0%B2%D0%B5%D1%81%D1%82> (дата обращения: 12.10.2016)
- 12.Принципы объектно-ориентированного программирования – Лекции по системе визуального объектно-ориентированного проектирования Delphi - Лекции (<http://blackman.wp-club.net/lection/visualprg.php>) (дата обращения: 11.10.2016)
- 13.Квест (значения) // Википедия [2015—2015] (дата обновления: 25.03.2015). - URL: <http://ru.wikipedia.org/?oldid=69566952> (дата обращения: 10.10.2016).
- 14.Международный вебинар "Живые" квесты в образовании" (современные образовательные технологии)" // Электронный журнал Экстернат.РФ URL: <http://ext.spb.ru/index.php/webinars/2209-22012013-qq-q-q.html> (дата обращения: 12.10.2016). Scratch и KODU: сходства и отличия URL: <https://edugalaxy.intel.ru/index.php?automodule=blog&showentry=5197> (дата обращения: 10.10.2016)
- 15.Kodu Game Lab Community URL: <http://www.kodugamelab.com/> (дата обращения: 11.10.2016)